REMARKS

Reconsideration and allowance of the subject application are respectfully requested.

In the independent claims 1 and 11, controlled access to data in a memory unit is performed on a domain basis. Secure data is accessible to the processor from the secure domain; it is not accessible from the non-secure domain. Non-secure data is accessible from the non-secure domain and may or may not be accessible from the secure domain. There can be multiple secure modes in the secure domain and multiple non-secure modes in the non-secure domain, athough the claims do not require multiple modes.

The memory unit stores both secure data and non-secure data. The allocation of data as either secure data or non-secure data is performed in the secure domain. An entry within the memory unit stores either secure or non-secure data at any particular point in time. A flag associated with each entry indicates whether that entry is secure data or non-secure data. Whenever the processor operates in a non-secure mode of the non-secure domain, the memory unit prevents access to any data item within an entry of the memory unit that the associated flag indicates has secure data stored therein. That secure data is only accessible from the secure domain.

The Examiner rejects all claims 1-20 under 35 U.S.C. §102(e) for anticipation based upon US-A-2003/0101322 (Gardner). This rejection is traversed.

To establish that a claim is anticipated, the Examiner must point out where each and every limitation in the claim is found in a single prior art reference. *Scripps Clinic & Research Found. v. Genentec, Inc.*, 927 F.2d 1565 (Fed. Cir. 1991). Every limitation contained in the claims must be present in the reference, and if even one limitation is missing from the reference,

then it does not anticipate the claim. *Kloster Speedsteel AB v. Crucible, Inc.*, 793 F.2d 1565 (Fed. Cir. 1986). Gardner fails to satisfy this rigorous standard.

Gardner describes a mechanism for protecting user application data so that it can be kept secret from root and other users (see Figure 6 and the associated description in paragraphs 0189 to 0193). There are four privilege levels PLO to PL3: PL0 is the most privileged level of the processor, and PL3 is the least privileged level of the processor (see paragraphs 0018 and 0019). The operating system runs at the privilege level PL2, and user applications run at the privilege level PL3 (see paragraph 0019). Figure 6 illustrates an approach where a user application operating at the least privileged level keeps its data secure from other users. Gardner refers to such user applications that require their data to be kept secure as "secure user applications." To accomplish that security, Gardner uses protection keys which allow a secure user application "to access a page of memory in memory 74 that nobody else can access, including root or anything running at PL2 or above" (see paragraph 0190). The protection keys are inserted into protection key registers by code executing at the protection level PLO (paragraph 0191).

Although Gardner keeps the data of secure user applications inaccessible by other user applications, Gardner does not teach controlling domain-based access to a memory unit. It is difficult to determine from the rejection, the specific elements of Gardner the Examiner is equating to particular features of the claims. At the bottom of page 3 and the top of page 4 of the Office Action, the Examiner appears to be equating various user processes and various privilege levels as analogous to modes of operation. With regard to the requirement for a plurality of domains, the Examiner refers to the phrase "secure and non-secure" appearing in paragraph 0189. But paragraph 0189 does not describe domains; instead, it relates to distinguishing secure user processes from non-secure user processes.

The operating system image layer can be partitioned into multiple independent protection domains which operate at privilege level PL2. The memory protection capabilities of the four privilege level processor hardware 32 protects those multiple independent protection domains running at PL2 from each other. Consequently, Gardner's protection domains are quite different to the claimed domains. In any event, Gardner does not control access to data in a memory unit on the basis of domains as recited in the independent claims: "wherein when the processor is in the secure domain, a program executed by the processor has access to secure data which is not accessible from the non-secure domain."

With respect to the claimed memory unit, the Examiner refers to elements 140 and 142 in Figure 3 of Gardner as being indicative of a plurality of entries. The page table 140 is a structure used to map virtual memory pages to physical memory pages. That mapping structure is not secure or non-secure data (see paragraph 0047). The independent claims recite that the memory unit entries store secure or non-secure data. Neither the page table 140 nor the virtual hash page table (VHPT) 142 themselves store any secure or non-secure data. The virtual hash page table (VHPT) is referred to when installing translation entries into the TLB 128 (see paragraph 0057).

Still further, Gardner fails to disclose associating a flag with each entry in the memory unit to indicate whether the one or more data items stored in the associated entry are secure data only accessible from the secure domain or non-secure data. The Examiner refers to paragraph 0189 as disclosing a flag, but this section merely describes the use of a magic number or ELF header to distinguish secure user applications from non-secure user applications. Distinguishing between applications has nothing to do with identifying data in the memory unit as being either secure data only accessible from the secure domain or non-secure data. Indeed, Gardner provides no details as to how data is stored within the memory unit. Nor does Gardner describe

using a flag associated with each entry.

Garnder also fails to disclose that the memory unit prevents access to any data item within an entry of the memory unit based on the value of the flag associated with that entry of the memory unit, as recited in the last paragraph of the independent claims. As mentioned by the Examiner, paragraph 0026 of Gardner teaches that the secure platform 40 ensures that one domain cannot accidentally or intentionally access another domain's memory. But the domains in Gardner are different from the domains claimed and described in the present application. In addition to this difference, paragraph 0026 teaches that it is the job of the secure platform 40 to control access to memory. In contrast, the last paragraph of the independent claims recite that it is the memory unit which controls access on the basis of the flags it has associated with each of its entries. No such technique is disclosed in Gardner.

The claimed memory unit is accessible from both the secure domain and the non-secure domain. But the integrity of secure data is ensured by making it accessible only from the secure domain. This is achieved by providing a flag associated with each entry in the memory unit that stores a value indicating whether the one or more data items stored in that associated entry are secure data or non-secure data. The memory unit itself is then able, when the processor is operating in a non-secure mode of the non-secure domain, to prevent access to any data item within an entry of the memory unit that the associated flag indicates has secure data stored therein. There is not even a hint of this approach in Gardner. Although Gardner is concerned generally with the issue of protecting data, Gardner is concerned with the very different problem of keeping the data of a particular user application secret from another user application, both user applications executing at the lowest privilege level (PL3) of the system. To achieve this, Gardner uses protection keys that are controlled by the PLO privilege level. This is a very

security approach than what is claimed.

Because Gardner lacks multiple features recited in the independent claims 1 and 11 and fails to address the particular problem solved by those claims, rejection based on Gardner is improper and should be withdrawn.

In addition to Gardner's deficiencies with respect to the independent claims, many of the specific bases for rejecting the dependent claim rejections are unfounded. Consider, for example, claim 2. One non-limiting application of the technology in this application is to a memory unit in the form of a cache. It is very useful for a cache to store both secure data and non-secure data and to be accessed from both the secure domain and the non-secure domain, while ensuring the integrity of the secure data. Without the approach recited in claims 1 and 2, it would typically be necessary to flush the contents of the cache when transitioning from the secure domain to the non-secure domain to ensure that there is no inadvertent access to secure data from the non-secure domain.

The Examiner refers to paragraph 0137 of Gardner, which merely describes the standard approach of arranging physical memory as a sequence of memory pages. Such physical memory is not a cache. Furthermore, nowhere in Gardner is there any discussion of the problem of maintaining the security of data held in a cache.

The application is in condition for allowance. An early notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:sd
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100